

Modern Systems Analysis, Inc.
1235 WINES DRIVE
ANN ARBOR, MICHIGAN 48103

Telephone (313)
761-7295

NTIS \$2.50

FINAL REPORT
for
CONTRACT NAS 9-12319

NEAR-EARTH ORBITAL GUIDANCE
AND REMOTE SENSING

TECHNICAL LIBRARY
SERIALS 43

(NASA-CR-128713) NEAR-EARTH ORBITAL
GUIDANCE AND REMOTE SENSING Final
Report (Modern Systems Analysis, Inc.,
Ann Arbor) 32 p HC \$3.75 CSCL 22B
N73-16876
Unclas
63/31 53547

DECEMBER 1972

9-12319-F

Final Report
for
Contract NAS 9-12319

Near-Earth Orbital Guidance
and Remote Sensing

by
William F. Powers

December 1972

Modern Systems Analysis, Inc.
1235 Wines Drive
Ann Arbor, Michigan 48103

TABLE OF CONTENTS

Section	Title	Page
1	Introduction	1
2	Remote Sensing	2
3	Parameter Optimization	4
3.1	The Parameter Optimization Algorithm . . .	4
3.2	Theoretical Basis	6
3.3	Space Shuttle Trajectory Optimization . .	12
4	Optimal Parameter Guidance	17
5	Conclusions and Recommendations	23
6	References	25
	Figures and Tables	27
	Figure 1. Flow Chart of the Algorithm. . .	27
	Table 1. Stage-And-Half Configuration; Good Initial x_0	28
	Table 2. Stage-And-Half Configuration; Poor Initial x_0	28
	Table 3. Ascent Problem with H-Matrix Tending to Singularity in DFP Method .	29

1. INTRODUCTION

Remote sensing and parameter optimization are two active space-related research areas. The purpose of this contract was twofold:

- (i) to present to the Mission Planning and Analysis Division a short course on the basics and state-of-the-art of remote sensing;
- (ii) to analyze recent developments in parameter optimization and determine their implications with respect to on-board guidance.

In Section 2, the format of the short course is outlined. In Section 3, results on parameter optimization are presented. The Fletcher and Broyden parameter optimization techniques are described and compared with the Davidon-Fletcher-Powell (DFP) method. Both of these methods have been built into the NASA-MSC PEACE parameter optimization program. In Section 4 the Johnson-Kamm parameter guidance scheme is discussed and a method for computing the feedback gains with a single numerical integration is presented. In Section 5 conclusions and recommendations for future study are presented, with an emphasis on the status and future of parameter optimization based on-board guidance.

2. REMOTE SENSING

A short course was presented during December 1971 through February 1972. The lecturers were members of the University of Michigan Willow Run and High Altitude Engineering Laboratories. The format of the course was as follows:

1st and 3rd Weeks: Basics of Remote Sensing and The User

Community. Fundamentals and definitions of remote sensing, user's needs (e.g., agronomists, resource managers, environmentalists, meteorologists, government agencies); basic mathematics and physics.

2nd Week: Multivariant Spectral Analysis. Relevant mathematics and physics review; retrieval of spectral, spatial, and temporal parameters; underlying principles of pattern recognition.

4th and 7th Weeks: Advanced Mathematics and Physics of Remote Sensing. Physics of spectroscopy, propagation, absorption, scattering, radiative transfer, geometrical and physical optics, and physics of sensing. Mathematics associated with the physics, basic probability and statistics, Fourier series and transforms, and numerical techniques.

5th Week: The Atmospheric Environment. Basic physics of the atmosphere, meteorological measurements from space vehicles, atmospheric parameters pertinent to environmental problems, circulation, sources and sinks for atmospheric pollutants.

6th Week: Imaging Sensing. Classification of passive and active sensors. Passive: multispectral scanners, infrared scanners.

Active: high-resolution radar. Characteristics of the sensors, data reduction, error sources, typical results.

8th Week: Nonimaging Sensing. Fourier transform spectroscopy, filter radiometry, microwave spectroscopy. Characteristics of the sensors, data reduction, error sources, typical results.

3. PARAMETER OPTIMIZATION

In Refs. 1-3, shuttle ascent trajectories are optimized by representing the steering angle rate as a sequence of straight line segments, and then applying the DFP parameter optimization technique. Eventually parameter optimization may be used on-board in the determination of steering commands. If so, then one should attempt to develop schemes which do not require an extensive 1-D search. One of the goals of this contract was to determine the relative performance capabilities of the DFP method and a modified Fletcher's method,^{4,5} which does not require an extensive 1-D search. In addition to this comparison, a relatively new method due to Broyden⁶ which requires a 1-D search was also considered since it is closely related to both the DFP and Fletcher methods. Both of these techniques have been built into the NASA-MSC PEACE program.

3.1 The Parameter Optimization Algorithm

Consider the minimization of the differentiable real-valued function

$$f(x_1, \dots, x_n). \quad (3.1)$$

All of the iteration schemes of this report utilize update formulas of the form

$$x^{(k+1)} = x^{(k)} - \alpha_k H_k g^{(k)}, \quad (3.2)$$

where $x^{(k)} \equiv$ current value of the vector x , $x^{(k+1)} \equiv$ new value of x , $\alpha_k \equiv$ a scalar parameter (the 1-D search parameter), $H_k \equiv$ an $n \times n$ matrix which is updated on each iterate, $g^{(k)} \equiv f_x(x^{(k)}) \equiv$ the gradient of f evaluated at $x^{(k)}$. A particular scheme

is defined by the way that it updates H_k and α_k . First the algorithm will be stated, and then the underlying theory will be discussed in Section 3.2. A flowchart of the scheme is given in Fig. 1. The notation $\Delta(\)_J \equiv (\)_{J+1} - (\)_J$ is employed below.

- (1) Specify x_0 , α_0 , H_0 , μ . Calculate $f_0 \equiv f[x_0]$, $g_0 \equiv g[x_0]$; set $J=0$. (H_0 is an arbitrary symmetric, positive definite matrix and $\alpha_0 > 0$.)
- (2) Calculate $f_{(J+1)1} \equiv f[x_J - \alpha_{J1} H_J g_J]$, where α_{J1} is the current estimate of α_J , and check $f_{(J+1)1} < f_J$. If yes, go to (3) if $\alpha_{J1} = 1$ or go to a crude step-size increase package if $\alpha_{J1} \neq 1$; if no, go to a crude step-size decrease package. The resultant step-size is denoted by α_J .
- (3) If $J=0$, go to (5). If $J>0$, check $\Delta f_J / (g_J^T \Delta x_J) \geq \mu$. If yes, go to (4); if no, decrease α_J until the inequality is satisfied and then go to (4).
- (4) Check $10^2 g_J^T \Delta x_J \leq g_{J-1}^T \Delta x_{J-1}$. If yes, go to (5); if no, increase α_J until the inequality is satisfied and then go to (5).
- (5) Calculate $x_{J+1} = x_J - \alpha_J H_J g_J$ and g_{J+1} . Check $\Delta g_J^T \Delta x_J > 0$. If yes, go to (6). If no, increase α_J .
- (6) Check $\Delta g_J^T \Delta x_J \geq \Delta g_J^T H_J \Delta g_J$. If yes, go to (8); if no, go to (7).

(7) Calculate:

$$H_{J+1} = H_J + \frac{\Delta X_J \Delta X_J^T}{\Delta X_J^T \Delta g_J} - \frac{H_J \Delta g_J \Delta g_J^T H_J}{\Delta g_J^T H_J \Delta g_J} \quad (3.3)$$

Go to (9).

(8) Calculate:

$$\begin{aligned} H_{J+1} = H_J - \frac{\Delta X_J \Delta g_J^T H_J}{\Delta X_J^T \Delta g_J} - \frac{H_J \Delta g_J \Delta X_J^T}{\Delta X_J^T \Delta g_J} \\ + \left(1 + \frac{\Delta g_J^T H_J \Delta g_J}{\Delta X_J^T \Delta g_J} \right) \left(\frac{\Delta X_J \Delta X_J^T}{\Delta X_J^T \Delta g_J} \right) \end{aligned} \quad (3.4)$$

Go to (9).

(9) If $\alpha_J < 1$, set $\alpha_{J+1} = \alpha_J$; otherwise, set $\alpha_{J+1} = 1$. Set $J = J+1$ and go to (2).

From the algorithm above one can see that $\alpha_J = 1$ is the desired value of the stepsize, and in the terminal iterations of the scheme $\alpha_J \rightarrow 1$ if the scheme is behaving like Newton's method as desired. Fletcher⁴ bases most of his discussion on the $\alpha_J = 1$ case and devotes little attention to the $\alpha_J \neq 1$ case. In the trajectory optimization problems of this report, the $\alpha_J \neq 1$ case occurs more often than not because of the difficulty of the problem and the use of finite difference formulas for the gradient calculations. Thus, more details about the $\alpha_J \neq 1$ case have been included in this section than in Ref. 4.

3.2 Theoretical Basis

The algorithm of Section 3.1 is basically a scheme for choosing between two formulas for the H_{J+1} -matrix while preserving a reasonable stepsize. Either Eq. (3.3) or (3.4) is

used to define H_{J+1} . Equation (3.3) is the classical DFP formula, a rank-two formula. Equation (3.4) is also a rank-two formula which has been studied in its own right in Refs. 6 and 7. The fact that Eq. (3.4) is rank-two may be seen by rewriting it as

$$H_{J+1} = \left(I - \frac{\Delta x_J \Delta g_J^T}{\Delta x_J^T \Delta g_J} \right) H_J \left(I - \frac{\Delta g_J \Delta x_J^T}{\Delta x_J^T \Delta g_J} \right) + \frac{\Delta x_J \Delta x_J^T}{\Delta x_J^T \Delta g_J} \quad (3.5)$$

In Ref. 6, Broyden shows that both Eqs. (3.3) and (3.4) are members of his one-parameter class of formulas introduced in Ref. 9, and that both satisfy the "quasi-Newton property", $H_{J+1} \Delta x_J = \Delta g_J$. Equation (3.3) results by choosing his β_J -parameter to be zero while Eq. (3.4) results if $\beta_J = 1/(\Delta g_J^T \Delta x_J)$. Broyden noted that in numerical experiments comparing the use of Eqs. (3.3) and (3.4) separately and with a 1-D search that the algorithms had similar characteristics in the early stages but quite different characteristics in the terminal stages. This behavior is explained by the fact that $\beta_J = 1/(\Delta g_J^T \Delta x_J)$ may be near zero in the early stages of the algorithm because the gradients may be relatively large (where $|\Delta g_J^T \Delta x_J| = |\varepsilon_{J+1}^T \Delta x_J - \varepsilon_J^T \Delta x_J| = |\varepsilon_J^T \Delta x_J|$ in a quasi-Newton scheme which employs a 1-D search). Since Eq. (3.4) and a 1-D search for the DFP program were required in the simulations, it was an easy task to also obtain simulations of Broyden's new method, i.e., Eq. (3.4) with a 1-D search.

Before considering Fletcher's justification for the basic algorithm, mention should be made of the occurrence of Eqs. (3.3)

and (3.4) in Ref. 7. Since numerous updating formulas for the H-matrix have been proposed in the past decade, Greenstadt⁸ considered the problem of choosing the "best" update formulas subject to appropriate constraints (e.g., symmetry and finite-convergence for a quadratic with a 1-D search). After investigating a number of performance indices, Greenstadt found that the following optimization problem gave tractable results:

$$\text{Minimize: } F(\Delta H_J) = \text{Tr}(W \Delta H_J W \Delta H_J^T) \quad (3.6)$$

$$\text{Subject to: } \Delta H_J^T = \Delta H_J \text{ (symmetry),} \quad (3.7)$$

$$\Delta H_J \Delta g_J = \Delta x_J - H_J \Delta g_J, \text{ (quasi-Newton)} \quad (3.8)$$

where $\text{Tr}(\) \equiv$ trace of $(\)$ and W is an arbitrary matrix to be specified. The expression obtained for ΔH_J by solving the above minimization problem involves the arbitrary matrix W . Goldfarb⁷ found that $W^{-1} = H_{J+1}$ results in Eq. (3.4) and $W^{-1} = H_{J+1} - (\Delta g_J^T \Delta x_J)(H_J \Delta g_J \Delta g_J^T H_J) / (\Delta g_J^T H_J \Delta g_J)^{3/2}$ results in Eq. (3.3), the DFP formula. He also showed that $z^T(\Delta H_J |_1 - \Delta H_J |_0)z \geq 0$, where $z \equiv$ arbitrary n-vector, $\Delta H_J |_1 \equiv H_{J+1} - H_J$ in Eq. (3.4), and $\Delta H_J |_0 \equiv H_{J+1} - H_J$ in Eq. (3.3). This means that Eq. (3.4) is less likely to tend toward singularity while Eq. (3.3) is less likely to tend toward unboundedness. Fletcher⁴ obtained a similar result by a different argument, and this forms the basis of his algorithm.

Let us now consider Fletcher's method.⁴ Denote the formula of Eq. (3.3) by H_0 and the formula of Eq. (3.4) by H_1 . Let ϕ be a scalar parameter and define the linear combination

$$H_\phi \equiv (1-\phi)H_0 + \phi H_1 \quad (3.9)$$

It is shown in Ref. 4 that if $\rho \in [0,1]$, then H_ρ possesses the following property: If $f(x)$ is a quadratic function with $G \equiv [f_{x_i x_j}]$ positive definite, then the eigenvalues of $G^{1/2} H_\rho G^{1/2}$ (arranged in order) tend monotonically to one for any sequence of vectors Δx . (I.e., H_ρ tends to the inverse Hessian G^{-1} in a certain sense.) Note that the property does not require a 1-D search. In addition to this property, it is shown that if $\rho \notin [0,1]$, then H^{-1} may diverge from G .

Since Eq. (3.9) represents an infinity of formulas, if it is to be useful there must exist a rule for selecting which value of $\rho \in [0,1]$ to use on a given iterate. Fletcher presents such a scheme by noting that a typical pitfall in the classical Davidon method is the tendency of the updating matrix H to become either singular or unbounded. He shows that if $\rho > \rho'$, then the eigenvalues $(\lambda_1, \dots, \lambda_n)$ of $H_\rho, H_{\rho'}$ (arranged in ascending order) are such that $\lambda_i(\rho) \geq \lambda_i(\rho')$ ($i = 1, \dots, n$), which implies $H_1 \equiv H_\rho \big|_{\rho=1}$ is "less singular" than $H_0 \equiv H_\rho \big|_{\rho=0}$, and H_0 is "less unbounded" than H_1 . Thus, a simple test for nearness to singularity would indicate whether to use H_0 or H_1 , which are the extreme elements of the class $H_\rho, \rho \in [0,1]$.

Fletcher shows that

$$\rho = \hat{\rho} \equiv \Delta g^T \Delta x / (\Delta g^T \Delta x - \Delta g^T H \Delta g)$$

defines the "rank one" formula

$$H_{J+1} = H_J + \frac{(\Delta x_J - H_J \Delta g_J)(\Delta x_J - H_J \Delta g_J)^T}{\Delta g_J^T (\Delta x_J - H_J \Delta g_J)} \quad (3.10)$$

The interesting thing about this formula is that if $\Delta g^T \Delta x > 0$, then $\hat{\rho} \notin [0,1]$, and the formula does not restrict the eigenvalues of H in any way. Thus, one can use the rank one formula to indicate which value of $\rho \in \{0,1\}$ should be used by simply checking the sign of $\Delta g^T \Delta x - \Delta g^T H \Delta g$; that is, if $\Delta g^T \Delta x > 0$ is enforced, then $\Delta g^T \Delta x - \Delta g^T H \Delta g > 0$ implies $\rho > 1$ (which means H_1 should be used) and $\Delta g^T \Delta x - \Delta g^T H \Delta g < 0$ implies $\rho < 0$ (which means H_0 should be used). If $\Delta g^T \Delta x - \Delta g^T H \Delta g = 0$, then H_1 is used to avoid singularity. Note that this test is step (6) of Section 3.1.

The only other steps in the algorithm which need to be discussed are steps (3) and (4). Step (3) is a check to determine if the stepsize is so large that an unreasonably small decrease in the function is attained. That is,

$$f_{J+1} = f_J + g_J^T \Delta x_J + O(\Delta^2) \quad (3.11)$$

implies

$$\Delta f_J / (g_J^T \Delta x_J) = 1 + O(\Delta). \quad (3.12)$$

If $0 < \Delta f_J / g_J^T \Delta x_J \ll 1$, then the decrease in cost is unreasonably small with respect to the steepness of the gradient.

Step (4) is a "filter" for the test

$$\Delta g_J^T \Delta x_J > 0. \quad (3.13)$$

It was noted in a number of simulations before the insertion of step (4) that condition (3.13) was violated. It is well known that if f is bounded from below, then there exists a larger value of α_J which will cause the inequality to be satisfied, and in Fletcher's paper a scheme for increasing α_J is presented. However,

this scheme might result in numerous costly gradient evaluations. (In the problems of the next section, a single gradient is approximately as costly as fourteen to eighteen function evaluations.) Since gradient calculations are so costly, an approximate test had to be devised to avoid the calculation of more than one gradient per iteration, and step (4) is the result.

It was noted that whenever the $\Delta g_J^T \Delta x_J > 0$ test was violated, the value of $|g_J^T \Delta x_J|$ was appreciably smaller than the value $|g_{J-1}^T \Delta x_{J-1}|$ (two to three orders of magnitude smaller). That is, on successive iterates on which $\Delta g^T \Delta x > 0$, the value of $g^T \Delta x$ was changing by zero to one-to-a-half orders of magnitude, whereas it changed by at least two to three orders of magnitude when the test was violated.

Note that to cause $|g_J^T \Delta x_J|$ to increase toward $|g_{J-1}^T \Delta x_{J-1}|$, one need only increase the value of the search parameter, which is the same remedy for the $\Delta g_J^T \Delta x_J > 0$ violation. Thus, the following test was employed before the computation of $g(x_{J+1})$:

$$100g_J^T \Delta x_J \leq g_{J-1}^T \Delta x_{J-1}, \quad (3.14)$$

i.e., if $g_J^T \Delta x_J$ is at least 100 times greater than $g_{J-1}^T \Delta x_{J-1}$, then the stepsize is increased and a decrease in $g_J^T \Delta x_J$ is guaranteed. (Note that $g^T \Delta x < 0$ is guaranteed on each iterate because of step (3).) For all the shuttle computations this test always detected the $\Delta g_J^T \Delta x_J > 0$ violation without computation of $g(x_{J+1})$ for an unacceptable x_{J+1} - value.

The test (3.14) has not been proved mathematically and it seems feasible that there exist cases when the test is satisfied

by $\Delta g_J^T \Delta x_J < 0$ and/or the tolerance value of 100 is unsuitable for other physical situations. However, $\Delta g_J^T \Delta x_J$ must be computed in each iteration for the H-formulas, and thus, the $\Delta g_J^T \Delta x_J > 0$ inequality can always be checked and guaranteed. In any case, no more computation is required than in the original Fletcher's method since $g_J^T \Delta x_J$ must be computed for other formulas in the method.

3.3 Space Shuttle Trajectory Optimization

A number of Space Shuttle trajectory optimization problems were simulated in the development of the algorithm, including three ascent problems and a reentry problem. A comparison of numerous algorithms for the stage-and-half configuration ascent problem are presented here along with partial results for a pressure-fed booster configuration ascent problem.

The stage-and-half optimization problem involved eighteen parameters (azimuth adjustment parameter, payload, pitch angle at the time when engines dropped, and fifteen pitch rates), where payload is to be maximized. The optimization is from ten seconds after liftoff to orbital insertion (50 x 100 with inclination specified). The results for this optimization problem are presented in Tables 1 and 2.

In Table 1 a comparison of DFP and the Modified Fletcher's method is shown for the case of a reasonably good guess for the initial parameter vector. The relatively small value of $g^T H g$ on the fifty-second iterate of the DFP method indicates that the problem is reasonably converged on that iterate. Considering the fifty-second iterate of the DFP method as the converged solution,

four digit accuracy is obtained by DFP on the twenty-sixth iterate and by the Modified Fletcher's Method on the thirty-first iterate (with respect to payload and boundary condition satisfaction). Since DFP requires more function evaluations and since a single gradient calculation corresponds roughly to eighteen function evaluations, the computing times to reach the twenty-sixth iterate in DFP and the thirty-first iterate in Modified Fletcher are approximately the same. As shown in Table 1, ten more iterates are obtained for the Modified Fletcher's method in the same amount of computer time. However, DFP gets a lower value for the cost in the same amount of computer time, thus exhibiting better terminal convergence.

In Table 2 a comparison of six algorithms is shown for a poor guess of the initial parameter vector. In the first column the gradient method (with a 1-D search) is included to show the difficulty of obtaining good terminal convergence in this problem. The next two methods, DFP and Broyden, were the best performers with Broyden slightly better than DFP. Note that DFP and Broyden give identical costs (to four digits) in the early iterates and then Broyden begins to get slightly lower costs; this is the same characteristic Broyden⁶ noticed. The last three columns show three methods which use only a crude search: Modified Fletcher, DFP with a crude search, and Broyden with a crude search. All three methods gave comparable results with Modified Fletcher obtaining the lowest cost in ten minutes computer time. All three H-formulas satisfy the main property of Fletcher's paper (i.e., $\lambda \in [0,1]$), and for this particular problem probably

give similar results because the H-matrix remains well-behaved. Although these three methods are not better than DFP or Broyden (with searches) on this problem, they are appreciably better than the gradient method and yet do not require extensive programming.

Table 3 shows an incomplete study of results obtained for the pressure-fed booster shuttle ascent problem. In this problem an element of the main diagonal of the H-matrix in the DFP method became appreciably smaller than the other elements of the main diagonal in the early iterates. This caused the 1-D search considerable trouble in obtaining a minimum, as noted by the large number of function evaluations (especially on the 3rd, 4th, and 5th iterates). In this particular problem the Modified Fletcher's method performed better than the DFP method in that it required a considerably less number of function evaluations and obtained a lower cost value in the same number of iterates. Note how the Modified Fletcher method uses both of the formulas on this problem (i.e., Eq. (3.3) is used 6 times, Eq. (3.4) is used 5 times).

During the course of the study a number of observations were made with respect to the performance of the algorithms and reports of their performance in the literature. These are summarized below.

- (1) The performance of the DFP method is strongly dependent upon the 1-D search used. In the early part of the study, the Modified Fletcher method required approximately the same amount of computer time as DFP to obtain the same cost on a

number of different problems. Then a more sophisticated search was used in the DFP algorithm. The DFP method then became a much better performer. This explains how, in the literature, numerous algorithms are reported to outperform DFP, when with an efficient search DFP is clearly the better performer. (In Ref. 4, Fletcher's method is reported to outperform DFP on a number of standard functions. However, when the two were compared with the NASA-MSC PEACE DFP program, DFP easily outperformed Fletcher's method.)

(2) In a number of papers in the literature, little emphasis is given to the expense of computing gradients as opposed to function evaluations. For example, the IBM Scientific Subroutine¹⁰ version of DFP calculates a gradient each time it evaluates the function. This calculation is not serious on low-dimension, test type problems, but it is extremely important when realistic problems are attacked (especially problems which require numerical integration for the function and gradient evaluations).

(3) In the early stages of the study, the effect of resetting to a gradient step every so many iterates was investigated. On the problems considered herein it was not found to be helpful; in fact, it was found to be detrimental in the terminal stages of convergence because the H-matrix had to be rebuilt. Most of the example problems in the literature which get improved convergence with reset are of relatively low-dimension. (One theoretical advantage of

reset is if it is included in any stable H-matrix type algorithm, then convergence can be proved for the same class of functions for which convergence can be proved for the gradient method.)

4. OPTIMAL PARAMETER GUIDANCE

Because of the success of the DFP algorithm in solving complicated Shuttle optimization problems, Ref. 11 proposes that the DFP algorithm may be useful for on-board guidance. A first approach to the problem is also presented in Ref. 11.

In this section we shall demonstrate the technique of Ref. 11 on a simple example, and reference some of the technical literature which is relevant to the areas of optimal neighboring and parameter guidance. Recommendations and conclusions are presented in Section 5.

Consider the following optimal control problem.

$$\text{Minimize: } J = \frac{1}{2}x_2(t_f)^2 \quad (4.1)$$

$$\begin{aligned} \text{Subject to: } \quad \dot{x}_1 &= u & x_1(0) &= x_2(0) = 0 \\ \dot{x}_2 &= \frac{1}{2}x_1^2 & |u| &\leq 1, \quad t_f = 2 \end{aligned} \quad (4.2)$$

The optimal control may be determined by inspection as

$$u(t) = \begin{cases} -1 & t \in [0, 1) \\ 0 & t \in (1, 2] \end{cases} \quad (4.3)$$

Since the optimal control is a sequence of piecewise constant segments, it satisfies the requirements of the Johnson-Kamm (J-K) guidance nominal control. The neighboring guidance function is

$$u(t) = \alpha_1[U(0) - U(1)] + \alpha_2 U(1), \quad (4.4)$$

where $U(a) = \begin{cases} 1 & t > a \\ 0 & t < a \end{cases}$ is the unit step function and α_1

α_2 are the constant controls on the intervals $[0, 1)$, $(1, 2]$, respectively. Of course, the values of α_1 and α_2 would be updated during each guidance cycle.

Suppose (t_0, x_{10}, x_{20}) , with $t_0 \in [0,1)$, is obtained by navigational measurements and (t_0, x_{10}, x_{20}) is near some point $(t, x_1^*(t), x_2^*(t))$ on the nominal trajectory. Given the control components α_1, α_2 (which we shall show how to calculate later), the final values of x_1 and x_2 are

$$x_1(2) = \alpha_1(1-t_0) + x_{10} + \alpha_2(2-1) \quad (4.5)$$

$$x_2(2) = x_{20} + \frac{1}{2} \int_{t_0}^1 [\alpha_1(t-t_0) + x_{10}]^2 dt + \frac{1}{2} \int_1^2 [\alpha_2(t-1) + x_{10} + \alpha_1(1-t_0)]^2 dt \quad (4.6)$$

Thus, the performance index is

$$J(\alpha; x_0) = \frac{1}{2} x_2(2)^2 = \frac{1}{2} \left\{ x_{20} + \frac{1}{2} \int_{t_0}^1 [\alpha_1(t-t_0) + x_{10}]^2 dt + \frac{1}{2} \int_1^2 [\alpha_2(t-1) + x_{10} + \alpha_1(1-t_0)]^2 dt \right\}^2 \quad (4.7)$$

Reference 11 suggests that this function of the guidance parameters should be minimized on the ground before the flight to determine nominal values for α_1^*, α_2^* (in this case, $\alpha_1^* = -1, \alpha_2^* = 0$), and then form the linear feedback guidance functions

$$\alpha_1 = \alpha_1^* + K_{11}(t_0)(x_{10} - x_1^*(t_0)) + K_{12}(t_0)(x_{20} - x_2^*(t_0)) \quad (4.8)$$

$$\alpha_2 = \alpha_2^* + K_{21}(t_0)(x_{10} - x_1^*(t_0)) + K_{22}(t_0)(x_{20} - x_2^*(t_0)),$$

where t_0 is the time of the navigational measurement with $t_0 \in [0,1)$. The nominal values $\alpha_1^*, \alpha_2^*, x_1^*(t), x_2^*(t), t \in [0,2]$ and time varying gains $K_{ij}(t), t \in [0,1), t \in (1,2]$ are stored on-board. As noted in Ref. 12, the choice of clock time, t_0 , as the reference time in Eq. (4.8) is usually nonoptimal. Procedures for determining the "best" index time are presented in

Ref. 12, e.g., the index time typically is the time on the reference trajectory which minimizes the state space distance between the measured state and nominal state trajectory.

The neighboring optimal feedback gains $K_{ij}(t)$ are determined by expending the first-order necessary conditions for a minimum of $J(\alpha_1, \alpha_2)$. That is, define

$$g_1(\alpha; x_0) = \frac{\partial J}{\partial \alpha_1}, \quad g_2(\alpha; x_0) = \frac{\partial J}{\partial \alpha_2}. \quad (4.9)$$

On the nominal trajectory

$$g_1(\alpha^*; x_0^*) = 0, \quad g_2(\alpha^*; x_0^*) = 0.$$

From Eq. (4.7), the expressions for g_1, g_2 are (assuming $t_0 \in [0, 1)$):

$$g_1 = \frac{\partial J}{\partial \alpha_1} = \left\{ \dots \right\} \left[\int_{t_0}^1 [\alpha_1(t-t_0) + x_{10}](t-t_0) dt + \int_1^2 [\alpha_2(t-1) + \alpha_1(1-t_0) + x_{10}](1-t_0) dt \right] \quad (4.10)$$

$$g_2 = \frac{\partial J}{\partial \alpha_2} = \left\{ \dots \right\} \left[\int_1^2 [\alpha_2(t-1) + \alpha_1(1-t_0) + x_{10}](t-1) dt \right] \quad (4.11)$$

where $\left\{ \dots \right\}$ is the bracketed term in Eq. (4.7).

Formally, the expansions of g_1 and g_2 about (α^*, x_0^*) are

$$g_i(\alpha; x_0) \approx g_i(\alpha^*; x_0^*) + \sum_{j=1}^2 \left[\frac{\partial g_i}{\partial \alpha_j} \Big|_* (\alpha_j - \alpha_j^*) + \frac{\partial g_i}{\partial x_{j0}} \Big|_* (x_{j0} - x_{j0}^*) \right] \quad (i=1, 2) \quad (4.12)$$

which implies (upon solution for α_1, α_2):

$$\begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix} = \begin{bmatrix} \alpha_1^* \\ \alpha_2^* \end{bmatrix} - \left[\frac{\partial g_i}{\partial \alpha_j} \Big|_* \right]^{-1} \left[\frac{\partial g_i}{\partial x_{j0}} \Big|_* \right] \begin{bmatrix} x_{10} - x_{10}^* \\ x_{20} - x_{20}^* \end{bmatrix} \quad (4.13)$$

The existence of finite feedback gains is dependent upon the existence of an inverse for $\left[\frac{\partial^2 \mathcal{L}}{\partial \alpha_j^2} \right]^*$. Define

$$P = \{ \dots \}, \quad (4.14)$$

where $\{ \dots \}$ is the bracketed term of Eq. (4.7). Then,

$$g_1 = P \frac{\partial P}{\partial \alpha_1}, \quad g_2 = P \frac{\partial P}{\partial \alpha_2}, \quad (4.15)$$

which implies

$$\left[\frac{\partial^2 \mathcal{L}}{\partial \alpha_j^2} \right]^* = \begin{bmatrix} P \frac{\partial^2 P}{\partial \alpha_1^2} + \left(\frac{\partial P}{\partial \alpha_1} \right)^2 & P \frac{\partial^2 P}{\partial \alpha_1 \partial \alpha_2} + \frac{\partial P}{\partial \alpha_1} \frac{\partial P}{\partial \alpha_2} \\ P \frac{\partial^2 P}{\partial \alpha_1 \partial \alpha_2} + \frac{\partial P}{\partial \alpha_1} \frac{\partial P}{\partial \alpha_2} & P \frac{\partial^2 P}{\partial \alpha_2^2} + \left(\frac{\partial P}{\partial \alpha_2} \right)^2 \end{bmatrix} \quad (4.16)$$

As $t \rightarrow 1$, $P \rightarrow 0$, $\frac{\partial P}{\partial \alpha_1} \rightarrow 0$ which implies the gains become infinite at the end of the constant control interval.

In general, as noted in Ref. 11, the gains become infinite at each constant control interval endpoint. This is analogous to infinite gains at the terminal time in existing guidance schemes. Since conservative allowances are made for the infinite-gain-at- t_f problem in existing guidance schemes, a similar procedure would be necessary for the guidance scheme of Ref. 11 at every constant control interval endpoint. Because of this the scheme would lose much of its optimality, and probably not be competitive with a physically based guidance scheme.

Another aspect of the scheme is that its normality or controllability properties are probably poorer than existing schemes. For example, in the example problem of this section,

if $t_0 \in (1, 2]$ then only one control parameter (α_2) is available. In a general problem where tight terminal boundary conditions are usually desired and the number of state variables is larger than the number of control variables (the usual case), the terminal phase would not have enough control parameters to influence all of the terminal state variables. Thus, a modified terminal phase would probably be required.

In Ref. 11 it is noted that a considerable amount of premission analysis would be required to compute the matrices required for the feedback gains because a forward integration of the differential equations for the matrices would be required from each guidance update time. Actually this computation can be reduced to a single backward integration by use of adjoint systems. That is, suppose

$$\dot{X} = A(t)X + B(t) \quad , \quad X(t_0) = X_0 \quad (4.17)$$

is the given matrix linear system. Define

$$Z^T = -Z^T A \quad (4.18)$$

Then,

$$\frac{d}{dt}(Z^T X) = Z^T B \quad (4.19)$$

or

$$Z^T(t_f^*)X(t_f^*) = Z^T(t_0)X(t_0) + \int_{t_0}^{t_f^*} Z^T(t)B(t)dt \quad (4.20)$$

Choosing

$$Z^T(t_f^*) = I, \quad (4.21)$$

and integrating (4.18) backward with the boundary condition (4.21) defines $Z(t)$. If the system is inhomogenous, then one more backward integration is required to define $\int_{t_0}^{t_f^*} Z^T(t)B(t)dt$. After these backward integrations, the matrix $X(t_f^*)$ for an

arbitrary $t_0 \in [t_0^*, t_f^*]$ is given by

$$X(t_f^*) = Z^T(t_0)X(t_0) + \int_{t_0}^{t_f^*} Z^T(t)B(t)dt. \quad (4.22)$$

To conclude this section mention should be made of the optimal parameter guidance scheme of Ref. 13. In Ref. 13, physically motivated parameters involved in analytical solutions for subarcs of the total trajectory are updated by use of the parameter conjugate gradient method on-board. Thus, use is made of an accelerated gradient technique on-board, and the conjugate gradient method was chosen because of less storage requirements with acceptable convergence properties. Even though the scheme of Ref. 11 was motivated by the excellent convergence properties of the DFP method in ground-based simulation, it does not employ DFP on-board. Furthermore, the only reason for associating the scheme with DFP is because the nominal trajectory is computed with DFP. More will be said about this aspect in Section 5.

5. CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusions

- 1.) The modified Fletcher parameter optimization scheme is an effective scheme for trajectory optimization which does not require an elaborate 1-D search. On well-behaved problems, DFP gave better performance, especially with respect to terminal convergence. The modified Fletcher method performed better on problems with ill-conditioned H-matrices.
- 2.) The Broyden method appears to be a promising new method. A related possibility is the Fletcher method with a 1-D search. All of these methods have been built into the NASA-MSD PEACE program.
- 3.) It does not appear that accelerated gradient methods have progressed to the point where they would be useful for on-board optimal parameter guidance. For such applications, they need to be improved in three main areas: (i) elimination or near-elimination of the 1-D search, (ii) guaranteed rapid terminal convergence, and (iii) reduction of storage requirements (the conjugate gradient method already satisfies this requirement). Since these areas are active research areas, improvements should be expected.
- 4.) The neighboring optimal parameter guidance scheme proposed in Ref. 11 does not appear to be feasible

because of infinite gain and controllability problems, in addition to the usual restrictions of a neighboring optimal guidance scheme.

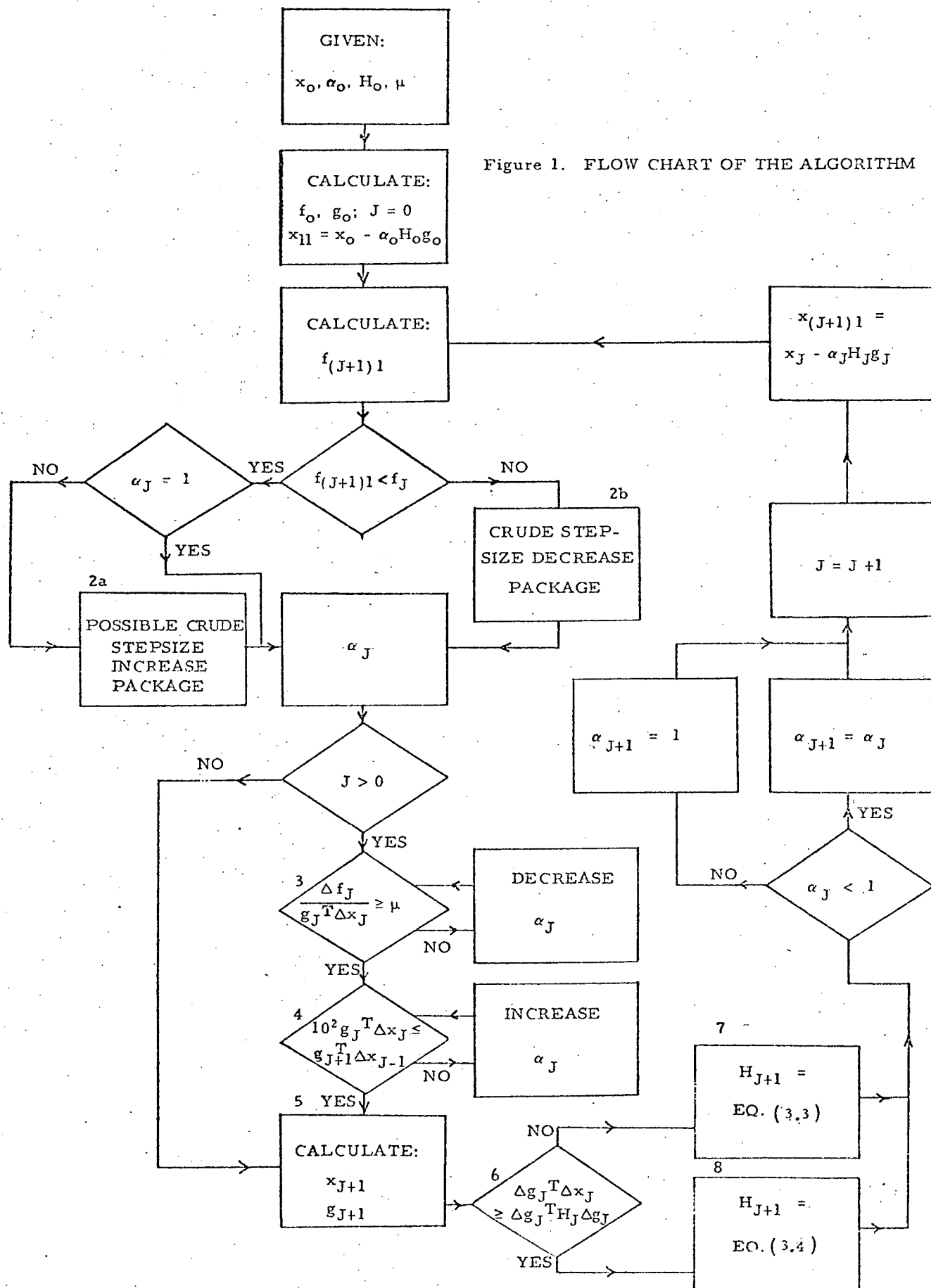
5.2 Recommendations

- 1.) Perform further simulations with the Broyden and Fletcher (with a 1-D search) parameter optimization schemes. They should be especially effective on problems where the H-matrix is ill-conditioned.
- 2.) Continue studying ways of eliminating the elaborate 1-D search from accelerated gradient parameter optimization schemes. This will hasten the use of these schemes in on-board guidance.
- 3.) The guidance scheme in Ref. 13 represents a reasonable blend of physical based guidance and on-board parameter optimization. It appears that this same idea may be applicable in reentry guidance. For example, the guidance scheme proposed in Ref. 14 is essentially a parameter guidance scheme with pieced subarcs of approximate analytical solutions. Thus, the possibility of adapting a parameter optimization scheme to the scheme should be investigated.

6. REFERENCES

1. Johnson, I.L. and Kamm, J.L., "Parameter Optimization and the Space Shuttle," Proceedings of the 1971 JACC, August 1971, pp. 776-781.
2. Johnson, I.L. and Kamm, J.L., "Near Optimal Shuttle Trajectories Using Accelerated Gradient Methods," AAS/AIAA Paper No. 328, August 1971.
3. Kamm, J.L. and Johnson, I.L., "Optimal Shuttle Trajectory-Vehicle Design Using Parameter Optimization," AAS/AIAA Paper No. 329, August 1971.
4. Fletcher, R., "A New Approach to Variable Metric Algorithms," Computer Journal, Vol. 13, No. 13, 1970, pp. 317-322.
5. Powers, W.F., "A Crude-Search Davidon-Type Technique With Application to Shuttle Optimization," AIAA Paper No. 72-907, September 1972.
6. Broyden, C.G., "The Convergence of a Class of Double-Rank Minimization Algorithms: 2. The New Algorithm," Journal of the Institute of Mathematics and its Applications, Vol. 6, No. 3, 1970, pp. 222-231.
7. Goldfarb, D., "A Family of Variable-Metric Methods Derived by Variational Means," Mathematics of Computation, Vol. 24, 1970, pp. 23-26.
8. Greenstadt, J., "Variations on Variable-Metric Methods," Mathematics of Computation, Vol. 24, 1970, pp. 1-22.
9. Broyden, C.G., "Quasi-Newton Methods and Their Application to Function Minimization," Mathematics of Computation, Vol. 21, 1967, pp. 368-381.
10. IBM System/360, Scientific Subroutine Package, (360A-CM-03X) Version III (1968), IBM Technical Publication Department, Page 221. (Corrected by IBM Memo, May 1969.)
11. Johnson, I.L., and Kamm, J.L., "On A Neighboring Optimal Guidance Scheme Using the Davidon Metric," NASA-MSc, Mission Planning and Analysis Division Internal Note, 1971.

12. Powers, W.F., "Techniques for Improved Convergence in Neighboring Optimum Guidance," AIAA J., Vol. 8, No. 12, December 1970, pp. 2235-2241.
13. Jacobson, R.A., and Powers, W.F., "Iterative Explicit Guidance for Low-Thrust Spacecraft," AIAA Paper No. 72-916, presented at 1972 AIAA Astrodynamics Conference, Palo Alto, California, September 1972.
14. Harpold, J.C., "Closed-Form Entry Guidance Logic for the High Cross-Range Orbiter," MSC Internal Note No. 72-FM-189, July 28, 1972.



Iteration Number	Davidon-Fletcher-Powell				Modified Fletcher		
	Cost	No. of Function Evaluations	Search Parameters	$\frac{df}{d\alpha} \Big _{\alpha=0} = -g^T H g$	Cost	No. of Function Evaluations	Search Parameters
0	$.13201 \times 10^1$	1	---	$-.79 \times 10^4$	$.13201 \times 10^1$	1	---
1	$.10640 \times 10^1$	12	$.646 \times 10^{-4}$	$-.90 \times 10^1$	$.10913 \times 10^1$	5*	$.125 \times 10^{-4}$
2	$.10322 \times 10^1$	4	$.701 \times 10^{-2}$	$-.68 \times 10^1$	$.10595 \times 10^1$	4*	$.312 \times 10^{-3}$
3	.98038	4	$.161 \times 10^{-1}$	$-.37 \times 10^1$	$.10480 \times 10^1$	5*	$.156 \times 10^{-2}$
4	.21987	6	.440	$-.28 \times 10^1$.57441	2*	.195
5	.14202	6	$.795 \times 10^{-1}$	$-.21 \times 10^0$.25564	2*	.195
10	$.84552 \times 10^{-1}$	6	$.124 \times 10^2$	$-.28 \times 10^{-4}$	$.85738 \times 10^{-1}$	4†	$.438 \times 10^{-1}$
20	$.84198 \times 10^{-1}$	4	$.876 \times 10^1$	$-.61 \times 10^{-6}$	$.84427 \times 10^{-1}$	2*	$.763 \times 10^{-2}$
30	$.84187 \times 10^{-1}$	3	.326	$-.36 \times 10^{-7}$	$.84212 \times 10^{-1}$	2*	$.477 \times 10^{-3}$
40	$.84182 \times 10^{-1}$	5	$.198 \times 10^3$	$-.48 \times 10^{-9}$	$.84206 \times 10^{-1}$	2†	$.298 \times 10^{-3}$
52	$.84181 \times 10^{-1}$	3	$.280 \times 10^1$	$-.79 \times 10^{-12}$	$.84204 \times 10^{-1}$	2*	$.596 \times 10^{-3}$
62	---	---	---	---	$.84202 \times 10^{-1}$	4*	$.142 \times 10^{-3}$

(43 minutes. Univac 1108; central differences)

*Used DFP formula (Eq. 3.3)

†Used Broyden formula (Eq. 3.4)

Table 1. Stage-And-Half Configuration; Good Initial x_0 .

Iteration Number	Gradient Method		DFP (with search)		Broyden (with search)		Modified Fletcher		DFP (crude search)		Broyden (crude search)	
	Cost	No. of Fn. Eval.	Cost	No. of Fn. Eval.	Cost	No. of Fn. Eval.	Cost	No. of Fn. Eval.	Cost	No. of Fn. Eval.	Cost	No. of Fn. Eval.
0	1611	1	1611	1	1611	1	1611	1	1611	1	1611	1
1	271.1	7	271.1	7	271.1	7	397.7	10*	397.7	10	397.7	10
2	54.65	5	13.75	4	13.75	4	34.64	5†	34.64	4	63.07	5
3	13.04	5	2.792	4	2.792	4	32.24	7*	32.27	7	30.72	6
4	10.83	4	1.311	4	1.311	4	30.95	3*	28.31	3	16.58	4
5	8.667	4	.8897	4	.8897	4	8.795	3†	7.948	3	7.395	3
6	7.736	4	.2636	5	.2636	5	6.003	3†	5.571	3	4.426	3
7	6.922	4	.1621	4	.1621	4	5.968	3†	4.809	3	3.422	3
8	6.438	4	.1563	4	.1563	4	3.123	3*	3.429	3	3.241	3
9	6.123	4	.1554	4	.1554	4	2.631	3†	2.850	3	2.388	4
10	6.038	4	.1461	4	.1461	4	2.420	3†	2.749	3	2.123	5
12	5.143	17	.1183	4	.1183	4	2.205	3*	1.815	4	.4776	3
14	3.574	17	.1046	4	.1045	4	.8923	4†	1.528	5	.4309	4
16	2.891	15	.0972	4	.0967	4	.5323	3*	.4339	5	.3682	4
18	2.834	17	.0923	5	.0921	5	.3663	3*	.3741	3	.3238	3
20	2.790	6	.0888	4	.0886	4	.3252	3†	.3619	3	.3104	3
22	2.718	4	.0875	5	.0873	5	.3092	3†	.3485	4	---	---
30	2.457	4	.08553	6	.08551	5	(10 min., Univac 1108; forward differences)					
50	1.739	25	.08516	6	.08508	6						
60	1.727	4	.08490	5	.08485	4						

(Approx. 40 min., Univac 1108; forward differences)

* Used DFP formula (Eq. 3.3)

†Used Broyden formula (Eq. 3.4)

Table 2. Stage-And-Half Configuration; Poor Initial x_0 .

Iteration Number	Davidon-Fletcher-Powell				Modified Fletcher		
	Cost	Number of Function Evaluations	Search Parameter	$\frac{df}{d\alpha} \Big _{\alpha=0} = -g^T H g$	Cost	Number of Function Evaluations	Search Parameter
0	1.058	1		-.8	1.058	1	---
1	1.058	4	$.25 \times 10^{-4}$	-.2	1.058	5*	$.12 \times 10^{-4}$
2	1.057	5	$.95 \times 10^{-2}$	$-.3 \times 10^{-1}$	1.057	7*	$.78 \times 10^{-2}$
3	1.056	26	$.35 \times 10^{-1}$	$-.9 \times 10^1$	1.057	3†	$.78 \times 10^{-2}$
4	1.056	37	$.68 \times 10^{-4}$	$-.1 \times 10^2$	1.056	6*	$.48 \times 10^{-3}$
5	1.053	30	$.10 \times 10^1$	$-.1 \times 10^{-1}$	1.056	11*	$.95 \times 10^{-6}$
6	1.050	5	.53	$-.4 \times 10^{-2}$	1.033	4*	$.24 \times 10^{-6}$
7	1.024	9	$.93 \times 10^1$	$-.4 \times 10^{-2}$	1.029	3*	$.12 \times 10^{-6}$
8	1.001	6	$.84 \times 10^1$	$-.3 \times 10^{-2}$.9816	3†	$.12 \times 10^{-6}$
9	.9883	5	$.68 \times 10^1$	$-.3 \times 10^{-2}$.9751	3†	$.12 \times 10^{-6}$
10	.9615	7	$.21 \times 10^2$	$-.3 \times 10^{-2}$.9607	3†	$.12 \times 10^{-6}$
11	.9592	7	$.16 \times 10^1$	$-.2 \times 10^{-3}$.9554	3†	$.12 \times 10^{-6}$
15	.9547	5	$.44 \times 10^1$	$-.1 \times 10^{-4}$	---	---	---
20	.9378	5	$.94 \times 10^1$	$-.6 \times 10^{-8}$	(7 min., Univac 1108; central differences)		
34	.9360	22	$.10 \times 10^8$	$-.2 \times 10^{-8}$			
	(22.5 min., Univac 1108; central differences)						

*Used DFP formula (Eq. (3.3)).

†Used Broyden formula (Eq. 3.4).

Table 3. Ascent Problem with H-Matrix Tending to Singularity in DFP Method.